American Journal of Computational Mathematics, 2014, 4, 183-196
Published Online June 2014 in SciRes. http://www.scirp.org/journal/ajcm
http://dx.doi.org/10.4236/ajcm.2014.43015

Scientific Research

Combining Symbolic Tools with Interval Analysis. An Application to Solve Robust Control Problems

Inès Ferrer-Mallorquí, Josep Vehí

Institut d'Informàtica i Aplicacions, Universitat de Girona, Girona, Spain
Email: ines@eia.udg.edu, vehi@eia.udg.edu

Received 19 February 2014; revised 19 March 2014; accepted 25 March 2014

Abstract

Complex systems are often subjected to uncertainties that make its model difficult, if not impossible to obtain. A quantitative model may be inadequate to represent the behavior of systems which require an explicit representation of imprecision and uncertainty. Assuming that the uncertainties are structured, these models can be handled with interval models in which the values of the parameters are allowed to vary within numeric intervals. Robust control uses such mathematical models to explicitly have uncertainty into account. Solving robust control problems, like finding the robust stability or designing a robust controller, involves hard symbolic and numeric computation. When interval models are used, it also involves interval computation. The main advantage using interval analysis is that it provides guaranteed solutions, but as drawback its use requires the interaction with multiple kinds of data. We present a methodology and a framework that combines symbolic and numeric computation with interval analysis to solve robust control problems.

Keywords

Interval Analysis, Robust Control, Symbolic Computation

1. Introduction

The description of real systems using mathematical models has been usually concerned with the concept of uncertainty. The inaccuracies and uncertainties can be done by different causes: in the linearization of a system, for the fact that some parameters of the system can have different values for different operation conditions or because some parameters can vary with time. Moreover uncertainties are classified into two types: parametric or non-parametric. In case the origin and the structure of the uncertainty were known, they are called structured or

How to cite this paper: Ferrer-Mallorquí, I. and Vehí, J. (2014) Combining Symbolic Tools with Interval Analysis. An Application to Solve Robust Control Problems. *American Journal of Computational Mathematics*, **4**, 183-196.
http://dx.doi.org/10.4236/ajcm.2014.43015

parametric, otherwise, if they are unknown, the uncertainty is classified as unstructured or non-parametric. From a theoretical point of view, there are a lot of approaches related with parametric robust problems. Most of them use sparametric methodologies [1]-[3]. Some using QFT [4], other Value Sets [5].

Some of these methods have been integrated into packages as Matlab [6] or into frameworks as PARADISE [7]. All the cited approaches, manage these problems of robust control using parametric methods. Although these methods have the drawback that the solutions obtained are no guaranteed solutions.

Methods for assessing robust stability and robust design of parameter dependent linear systems, as Routh table or the small gain theorem,... give sufficient (but not always necessary) conditions to solve them. These theorems give a powerful methodology to transform robust control problems in simplified problems consisting of testing the positivity of a set of functions or conditions.

This paper presents a methodology that solves robust control problems combining symbolic computation with interval analysis [8] techniques. The main advantage of interval analysis is that it offers guaranteed solutions in case we want to solve robust control problems. Thus, its use means a really improvement in front parametric techniques.

Although the advantage of obtain guaranteed results is important, the rules and algorithms that must be generated to solve robust control problems using interval analysis, are difficult to be implemented for users. This difficulty discourage control engineers to use these techniques. This difficulty motivates us to offer the users an automatic application that uses interval techniques in a transparent way for the user. This application or framework makes a combination between symbolic tools and interval analysis methods. This paper presented his framework named IRCAD (Interval Robust Control framework for Analysis and Design) dedicated to the resolution of robust control problems applying techniques of interval analysis. This framework is composed by a set of tools of analysis and design that combine symbolic computation with the methodology of interval analysis. Thus, it is not necessary that the user have any special knowledge about interval analysis to use the functions of robust analysis and robust design that offer the framework.

The paper is organized as follows. Section 2 presents the formulation of the most typical robust parametric control problems. The use of interval analysis theory as a way to obtain guaranteed results on the solution of robust parametric control problems is the central point of Section 3. The problem of how to integrate all this diversity of data and all the involved methodologies of this kind of problems is discussed on Section 4. In Section 5 IRCAD is presented as a framework that allows the integration of the different data. Section 6 summarizes the framework and applies it to an illustrative example. Finally most emphasizing points and future intentions are put on the Section 7 of conclusions.

2. Robust Parametric Control Problem

To formulate the problem of parametric robust control we consider a system that is linear respect the variable s. This kind of systems can be expressed as a transfer function or in space state format, in continuous or in discrete domain.

In this paper we will consider a class of plants [9] with structured parametric uncertainties described by the following uncertain transfer function:

$$G(s,q)=\frac{\alpha_0(q)+\propto_1(q)s+\cdots+\alpha_m(q)s^m}{\beta_0(q)+\beta_1(q)s+\cdots+\beta_n(q)s^n} \tag{1}$$

depending on a structured perturbation characterized by the parameter vector

$$\boldsymbol{q}=\begin{bmatrix}q_1 & q_2 & \cdots & q_l\end{bmatrix}^{\mathrm{T}} \tag{2}$$

where each parameter enters into the system description with polynomial dependency. We also consider a certain configuration of the feedback system with a fixed controller $C(s,\boldsymbol{k})$, where $\boldsymbol{k}$ is the design parameter vector. The motivation of this kind of system description is that the system parameters, $\boldsymbol{q}$, can represent physical quantities that are known only to within a certain accuracy, or vary depending on operating conditions while the controller parameters, $\boldsymbol{k}$, represent degrees of freedom available to the control system designer. Due to the physical interpretation of the uncertain parameters, each one can be considered independent from the other and their values lie between upper and lower bounds. Then, the uncertain domain can be defined as any hyperrectangle:

$$\boldsymbol{Q}=\left\{\boldsymbol{q}=\begin{bmatrix}q_1 & q_2 & \cdots & q_l\end{bmatrix}^{\mathrm{T}}\middle| q_i\in\left[q_i^-,q_i^+\right],\ i=1,\cdots,l\right\} \tag{3}$$

With this kind of feedback system description, several robust control problems can be formulated. Problems concerning robust control analysis as testing the stability of the system or computing the stability margin. Also, problems related with control design as to find a set of controllers that fulfill one or more than one selected specifications.

2.1. Robust Control Problems: Analysis

In many control systems the plant parameters may vary over a wide range about a nominal value $\boldsymbol{p}^0$. Robust parametric stability refers to the ability of a control system to maintain stability despite such large variations. Given an uncertain system and a controller, analysis of robust control problems consists in to check(calculate) some performance specifications:

- **Robust stability**

The first problem formulated is the verification of the stability on uncertain systems. That is, how to find the necessary and sufficient conditions that allows to guarantee that a family of uncertain polynomial is stable or no. In [10], Ackermann presents different methods to check robust stability analysis.

-*Boundary crossing theorem*: It is based on Hurwitz determinants. Given a set of polynomials $P(s,\boldsymbol{Q})$, this set is robustly stable, if and only if:

1) There exists a stable polynomial $p(s,\boldsymbol{q})\in P(s,\boldsymbol{Q})$,
2) Any root cross imaginary edge: $jw\notin \text{Roots}\left[P(s,\boldsymbol{Q})\right]$ for all $\omega\geq 0$

-*Alternative boundary crossing theorem based on Hurwitz Determinants* [11]: Giving a set of polynomials $P(s,\boldsymbol{Q})$ it is robustly stable, if and only if:

1) There exists a stable polynomial $p(s,\boldsymbol{q})\in P(s,\boldsymbol{Q})$
2) $\det H_n(\boldsymbol{q})\neq 0$ for all $\boldsymbol{q}\in\boldsymbol{Q}$

This work uses the just presented boundary crossing theorem in a modified way. It gives necessary and sufficient conditions of absolute robust stability for a family of characteristic polynomials.

- **Stability margin**

If the controller is given, the parametric stability margin is defined [12] as the maximal range of variation of the parameter $\boldsymbol{p}$, measured in a suitable norm, for which closed-loop stability is preserved. It can be expressed as:

$$\rho_x := \sup\left\{\alpha:\delta(s,\boldsymbol{x},\boldsymbol{p})\ \text{stable},\left\|\boldsymbol{p}-\boldsymbol{p}^0\right\|<\alpha\right\} \tag{4}$$

This margin is used as a quantitative measure of the robustness of the closed loop system with respect to parametric uncertainty evaluated at the nominal point p^0.

- **Parametric bode plots**

Based on the experience with frequency domain analysis methods, one might conjecture that it is not necessary to check all $\boldsymbol{\omega}$ values. Thus, it is common to construct Bode plots using only a grid of $\boldsymbol{\omega}$ values. In the case of frequency domain methods for robustness analysis, the bottleneck is "singular frequencies".

In order to provide a systematic method of identifying singular frequencies, an algebraic definition is given on [10]. Before the definition some preliminary notation is introduced. When an uncertain polynomial $p(s,\boldsymbol{q})$ is evaluated at a frequency $\boldsymbol{\omega}$ it equals a complex number $p(j\omega,\boldsymbol{q})=h\left(-\omega^2,\boldsymbol{q}\right)+j\omega g\left(\omega^2,q\right)$. Separating this complex number on real and imaginary parts:

$$h\left(-\omega^2,\boldsymbol{q}\right)=a_0(\boldsymbol{q})-a_2(\boldsymbol{q})\omega^2+a_4(\boldsymbol{q})\omega^4-\cdots \tag{5}$$

$$\omega g\left(-\omega^2,q\right)=a_1(q)\omega-a_3(q)\omega^3+a_5(q)\omega^5-\cdots \tag{6}$$

Associated with these functions is a Jacobi an matrix.

$$J(\omega,\boldsymbol{q})=\begin{pmatrix}\dfrac{\delta h\left(-\omega^2,\boldsymbol{q}\right)}{\delta q_1} & \dfrac{\delta h\left(-\omega^2,\boldsymbol{q}\right)}{\delta q_2} & \cdots & \dfrac{\delta h\left(-\omega^2,\boldsymbol{q}\right)}{\delta q_l}\\ \dfrac{\delta\omega g\left(-\omega^2,\boldsymbol{q}\right)}{\delta q_1} & \dfrac{\delta\omega g\left(-\omega^2,\boldsymbol{q}\right)}{\delta q_2} & \cdots & \dfrac{\delta\omega g\left(-\omega^2,\boldsymbol{q}\right)}{\delta q_l}\end{pmatrix}$$

In the Jacobian $J(\omega,\boldsymbol{q})$, the real part $h(-\omega^2,\boldsymbol{q})$, and the imaginary part $\omega g(\omega^2,q)$ are used to give algebraic definition of singular frequencies.

Definition. The nonnegative frequency ω_s is a singular frequency of the uncertain polynomial $p(s,\boldsymbol{q})$ if there exists a $\boldsymbol{q}^0 \in \mathbb{R}^l$ such that the three following conditions are simultaneously satisfied.

$$h(-\omega_s^2,\boldsymbol{q}^0)=0 \tag{7}$$

$$\omega_s g(-\omega_s^2,\boldsymbol{q}^0)=0 \tag{8}$$

$$rank\left[\boldsymbol{J}(\omega_s,\boldsymbol{q}^0)\right]<2 \tag{9}$$

The last Equation (9) can be substituted by: $\det \boldsymbol{J}(\omega\boldsymbol{q})=0$

- **Frequency problems**

-*Value sets*. Some gridding approaches have as a drawback the length of time it takes to compute the set of frequency plots. An alternative and faster technique, suggested in [5], is to compute the frequency plot $p(j\omega,\boldsymbol{q}),\omega\geq 0,$ for each $\boldsymbol{q}$ on a grid of Q. It is advisable to compute the *value set* for each ω on a grid of frequencies from 0 to +∞.

$$P(j\omega,Q)=\left\{p(j\omega,\boldsymbol{q})\in\mathbb{C}\middle|\boldsymbol{q}\in Q\right\} \tag{10}$$

In the case of a family of polynomials, the stability test is based on repeating the construction of the value set for a grid of frequencies to give the set of all possible frequency plots. The collection of value sets would then indicate stability or instability depending on how the zero-exclusion theorem is formulated.

Theorem 1. (*zero-exclusion theorem*). Given a polynomial family $P(s,Q)=\left\{p(s,q)\middle|q\in Q\right\}$. This set is robustly stable if and only if

- A stable polynomial $p(s,\boldsymbol{q})\in P(s,Q)$ exists and
- $0\notin P(j\omega,Q)$ for all $\omega\geq 0$.

- **Discrete-time problems**

Some problems concerned with discrete-time are:

-*Model conversion*. Conversion is needed in both senses, from a continuous time interval state-space model to a discrete-time interval model and also from a discrete-time uncertain system to a continuous-time uncertain model.

-*Schur stability test*. The problem of checking the stability of a discrete-time system is reduced to the determination of whether or not the roots of the characteristic polynomial of the system lie strictly within the unit disc, that is whether or not the characteristic polynomial is a Schur polynomial. If

$$P(z)=p_n z^n+p_{n-1}z^{n-1}+\cdots+p_1 z+p, \tag{11}$$

where the z_i are the *n* roots of $P(z)$. Then if $P(z)$ is Schur, all these roots are located inside the unit circle $|z|<1$, so that when z varies along the unit circle $z=\mathrm{e}^{j\theta}$, the argument of $P(\mathrm{e}^{j\theta})$ increases monotonically. For a Schur polynomial of degree *n*, $P(\mathrm{e}^{j\theta})$ has a net increase of argument of $2\pi n$, and thus the plot of $P(\mathrm{e}^{j\theta})$ encircles the origin *n* times. This can be used as frequency domain test for Schur stability [13] [14].

-*Non-linear discrete-time control of uncertain systems*. This problem can be formulated as:

Find one c

$$c\in S_c=\left\{c\in C\middle|\forall p\in P, f(c,p)>0\right\} \tag{12}$$

where *f* is a vector function that can be evaluated using algorithms based on interval analysis. This problem is both quite complex because it involves a quantifier and, at the same time, very simple because the only objective is to find one feasible vector. This makes considering a larger number of tuning parameters corresponding to the guaranteed-tuning problem [15]. This is evidence that it is possible to combine non-linearity, and structured uncertainty with guaranteed results.

2.2. Robust Control Problems Design

The requirement of robust design is that the design specifications must be satisfied over the entire parameter set. Given an uncertain system, the problem is to obtain a robust controller that fulfills some performance specifica-

tions.

As robust performance specifications we can enumerate stability, pole locations, steady-state, H_∞, etc. It is known that the solution of these problems can be reduced to the problem of checking the positivity of a set of rational functions over a given domain:

$$f_i(\alpha, \boldsymbol{q}, \boldsymbol{k}) > 0 \quad \forall \alpha \in A \quad \forall \boldsymbol{q} \in \boldsymbol{Q} \quad \forall \boldsymbol{k} \in \boldsymbol{K} \tag{13}$$

where α is the generalized frequency and $\boldsymbol{k} \in \boldsymbol{K}$ may be a single point (nominal controller k^0) or a certain domain in the parameter's space of the controller depending on the problem considered.

- Robustness analysis: to check if the controlled system achieves robust performances.

$$f_i(\alpha, \boldsymbol{q}) > 0 \quad \forall \alpha \in A \quad \forall \boldsymbol{q} \in \boldsymbol{Q} \tag{14}$$

- Robust control design: to find the robust K set such that the controlled system achieves robust performances.

$$f_i(\alpha, \boldsymbol{q}, \boldsymbol{k}) > 0 \quad \forall \boldsymbol{k} \in \boldsymbol{K} \quad \forall \boldsymbol{q} \in \boldsymbol{Q} \tag{15}$$

Most of the more used robust control design techniques are based on the worst possible scenario which may never occur in a particular control system. Some of these techniques are the following:

- $\boldsymbol{H_\infty}$ **control** [16]. Robust performance specifications may be given as pole locations as well as H_∞ performances expressed in the frequency domain.
- $\boldsymbol{l_1}$ **control** [17]. In the standard l_1 problem the design of an internally stabilizing controller such that the l_1 norm of the regulated output z due to the worst-case magnitude bounded disturbance ω is addressed. The Banach space of right-sided absolutely summable real sequences with the norm given by

$$\|x\|_1 := \sum_{k=0}^{\infty} |x(k)| \tag{16}$$

- $\boldsymbol{\mu}$ **synthesis**. It minimizes the superior bound of the robust performance. The criterion is

$$J = \min_{t_p} \left\| \left[\min_{c(s)} \left\{ \sup \mu_{A(s)} \left[M(s) \right] \right\} \right] - 1 \right| \tag{17}$$

where $\mu[0]$ is the structured singular value, $A(s)$ is the disturbance matrix which indicates that $\mu[0]$ depends on the allowed structure for $A(s)$, $M(s)$ is a matrix obtained by rearranging the uncertain system into the M A-structure and $C(s)$ indicates that the minimisation is function of the controller parameters [18] [19].

- QFT (*Quantitative feedback theory*). While value sets are mainly concerned with analysis, when the aim is finding a compensator to satisfy design specifications, QFT is the most important approach. It can be considered as a natural extension of classical frequency-domain design approaches. One of the main objectives is to design a simple low-order controller where the bandwidth of the feedback controller being as small as possible. At a fixed frequency, the plant's frequency response set is called a template. In the bound generation step of QFT design procedure, the plant template is used to translate the given robustness specifications in domains in the Nichols chart where the controller gain-phase values are allowed to lie.

We are focused on the solution of robust control design problems. That is, we are testing if a controller *K* achieves the design specifications.

Some applications can be found in the design field. Parameter spaces of the controller are explored, finding the regions which fulfill a control specification in order to obtain a controller which satisfies it.

3. Interval Formulation of Robust Parametric Control

The problem of robust parametric control, presented on the last section, is now formulated using interval analysis. Using the theory of interval analysis is possible to obtain guaranteed solutions to robust control problems. To formulate these problems on interval format, we formulate the problem of the parameter space approach considering the uncertain system of **Figure 1**, where $\boldsymbol{k}$ is the parameter vector of the controller,

$$\boldsymbol{k} = \begin{bmatrix} k_1 & k_2 & \cdots & k_l \end{bmatrix}^{\mathrm{T}} \tag{18}$$

$\boldsymbol{q}$ is the parameter vector of the process, and the uncertainty domain can be defined as a box:

$$K = \left\{ \boldsymbol{k} = \begin{bmatrix} k_1 & k_2 & \cdots & k_l \end{bmatrix}^{\mathrm{T}} \middle| k_i \in \left[\underline{k_i}, \overline{k_i} \right] \right\} \tag{19}$$

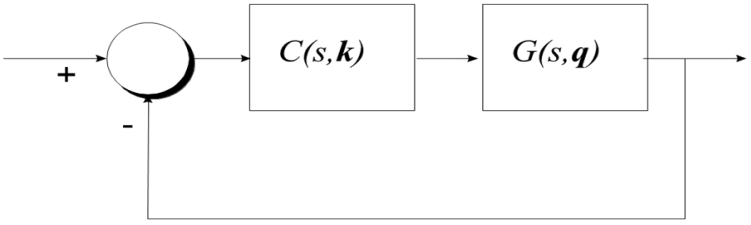

Figure 1. Uncertain system.

Design specifications are formulated in terms of closed-loop system stability and performances in the frequency domain. These specifications, such as bandwidth, resonance peak, control effort, etc., can be described as a set of *N* inequalities of the type:

$$f_i(\omega, \boldsymbol{q}, \boldsymbol{k}) > 0 \quad \omega \in \Omega \quad \boldsymbol{q} \in \boldsymbol{Q} \quad \boldsymbol{k} \in \boldsymbol{K} \quad i = 1, \cdots, N \tag{20}$$

where Ω is a subset of $\mathbb{R}^+$ (usually an interval), $\boldsymbol{Q}$ an interval vector representing interval parameters and $\boldsymbol{K}$; a nondegenerate set.

Then robust design can be formulated as follows: given a controller structure, $C(s, \boldsymbol{K})$ the aim is to find the values of $\boldsymbol{k}$ which conform with the robust control specifications. Taking this formulation into account some control problems can be proposed [20]:

1) *Performance checking*. Given the uncertain system $G(s, \boldsymbol{Q})$ and the uncertain domain $\boldsymbol{Q}$, the problem is to test if the designed controller $C(s, \boldsymbol{k}^0)$ achieve the robustness specifications obtained from Equation (20):

$$F_i(\omega, \boldsymbol{Q}, \boldsymbol{k}^0) > 0 \tag{21}$$

2) *Performance margin computation*. Taking set $\prod$ as a function of its radius ρ:

$$\prod(\rho) = \left\{ \boldsymbol{q} : \left\| \boldsymbol{q} - \boldsymbol{q}^0 \right\|_\infty^\omega \le \rho \right\} \tag{22}$$

find the maximal set $\prod(\rho^*)$ so that the designed controller $C(s, \boldsymbol{k}^0)$ achieves robust performances for all ρ belonging to $\prod(\rho^*)$.

3) *Robust controller design*. Taking a particular structure for the controller and specifying the uncertain domain where the parameters of the system $\boldsymbol{Q}$ can vary, find a fixed structure controller $C(s, \boldsymbol{k}^0)$ so the controlled closed loop system achieves the robust performance specifications:

$$F_i(\omega, \boldsymbol{Q}, \boldsymbol{k}^0) > 0 \tag{23}$$

4) *Obtaining the set* $\boldsymbol{K}$ *for the robust controller problem*. Given an uncertain plant $G(s, \boldsymbol{q})$ the variation domain of the system parameter $\boldsymbol{Q}$ and a controller structure, find the robust set $\boldsymbol{K}$ which allows $C(s, \boldsymbol{k})$ to achieve the robustness specifications:

$$F_i(\omega, \boldsymbol{Q}, \boldsymbol{k}) > 0 \tag{24}$$

5) *Estimating the stability region problem, for a given* $\boldsymbol{k}^0$. This problem consists on building a set of all ρ 's achieves closed loop stability when given $\boldsymbol{k}^0$.

In our approach two robust control problems, Design and Analysis, have been converted into a set of functions using symbolic computation. These functions have been intervalised resulting on a set of interval functions. At this point, to solve the problems it has only been necessary to test the positivity of these interval functions for all the variation of its parameters, Equation (14) and Equation (15).

4. Integration of Numeric, Interval and Symbolic Computation

As it has seen in Section 2, problems related with robust parametric control systems use symbolic data due these systems involves uncertain variables. Moreover in Section 3 is denoted that the interval formulation of the problems involves numeric and interval data. Thus, solving robust control problems using interval analysis tools, offer guaranteed results, but as a drawback, its use requires the interaction with multiple kind of data. These difficulties can discourage control engineers to use interval techniques although its benefits.

To make all these tools and techniques more available is interesting an environment that integrates the treatment of all these kind of data, allowing deal with robust control problems, solving them using a common methodology.

In this sense our work has found how to offer to the control engineer a set of friendly user interface tools that integrates the diversity of data that these kinds of problems involves and also allows to use interval tools in a transparent way for the user.

In **Figure 2**, a schema of the different tools used for these kinds of problems to manage numeric symbolic and interval computation is shown.

As it can be seen in this schema, Matlab is the package used to centralize our environment. It is the selected package to develop the most routines of our environment due three main points:

- The high computational power with problems that involves matrix data.
- Its facility to deal with high level languages.
- Its easy connection with other commercial packages as Maple.

Our framework solves robust control problems, thus, problems that have implicitly high level degree of computation. This aspect would be a bottleneck if they would be solved writing scripts on a package like Matlab. Due this high cost of computation the routines on this framework have been implemented with the high level language C++. This language improves the building of procedures which require numeric-interval computation and allows to execute code in a faster way than using Matlab. This achievement was very decisive. In addition it allows to use tools from interval analysis libraries [21], due its include instructions.

Once presented the two main packages used by the framework, it rests to say how Matlab procedures link with C++ critical routines. The deal of procedures and data among these two packages is done using code CMEX and DLL functions. The high level C++ functions are compiled using Borland, obtaining a set of dll function. There are executable functions that can be called from Matlab commands or from Matlab scripts.

Another aspect to consider is how to enter symbolic data, such as transfer functions or problem specifications. In our case it has been choose the Maple package. The extraction of these symbolic inputs from Matlab scripts and the pass of them to the Maple Package are implemented using the Symbolic Math Toolbox.

Therefore, Matlab is used to make the integration of numeric, interval and symbolic data, taking symbolic inputs, transforming them into a suitable format for the C++ routines, passing these transformed inputs to the C++ routines and finally collecting the data returned.

The last step, shown in the lower part of the diagram (user interface block) corresponds to the need to show results in a suitable way for control applications. The framework has implemented this part using the GUI-tools of Matlab. These tools allow to achieve a friendly interface that allows to present the results in an adequate format to be analyzed easily by the user.

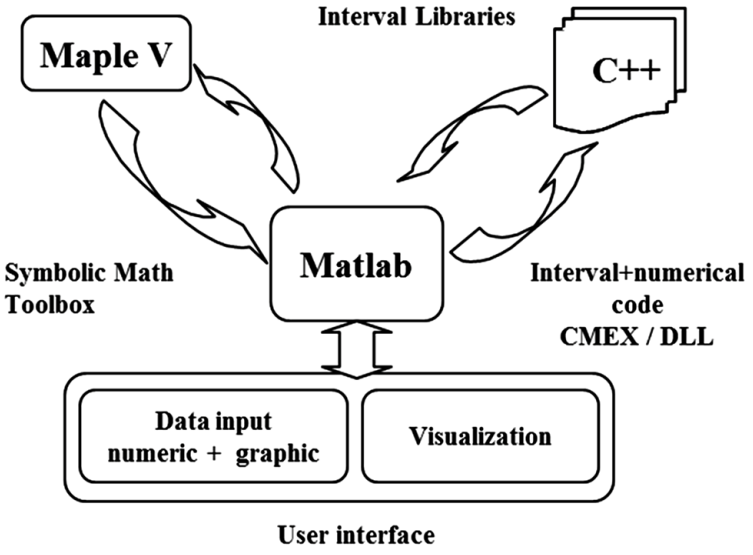

Figure 2. Cooperation numeric-interval-symbolic.

5. IRCAD: A Framework for Robust Control Problems

Once introduced the need to integrate the diversity of data and methods in section 4, now in this section we present the developed environment in order to achieve this integration. To obtain the kindly manipulation of numeric, interval and symbolic computation, our work has built a set of tools under the format of a framework called IRCAD (Interval Robust Control framework for Analysis and Design). To introduce the framework IRCAD (main screen is showed on **Figure 3**), we present the functions that it integrates. These functions or tools could be laid as three sets of problems.

The first set includes robust analysis problems like to test the stability of a system and to find the stability margin. The second set of problems are related to robust design problems and the main tool is an application that allows to select the design specifications that we want our control system achieve. The framework includes a third set corresponding to post-design tools, which allows that a selected controller can be applied to the system without leave the framework IRCAD.

The starting point to use the tools of the framework is to input the transfer function of the plant. Due the systems we consider are uncertain systems, the plant is defined as a family of transfer functions. To input them into the framework it is necessary to define a transfer function with undefined parameters, and also define their uncertain values range (interval values). So, as example, we can define a transfer function for a family of plants like $G(s,\boldsymbol{q})$ with the uncertain parameter $\boldsymbol{q}=[q_1,q_2]$.

5.1. Tools for Robust Analysis

The framework IRCAD has a set of tools addressed to solve robust analysis problems. Giving an uncertain system, where the transfer function is defined as a family of plants q_i, from the menu Analysis, it is possible to make the following functions:

-*Stability test*. To calculate the absolute stability region, a branch-and-bound algorithm based on interval analysis is used. The result of this tool shows the regions stable and unstable. In case the depth of the algorithms was limited (in order to accelerate the end of the computation) can appear undefined regions. In the case there aren't problems of time computation, due the computer has high computation resources, it is possible to don't limit the depth of search and then the lonely regions that appear are estable and unestable regions.

-*Stability margin computation*. The parametric stability margin is defined as the length of the smallest perturbation Δq which destabilizes the closed loop. This margin is used as a quantitative measure of the robustness of the closed loop system with respect to parametric uncertainty evaluated at the nominal point q_0.

-*Parametric Bode plots*. It is common to construct Bode plots using only a grid of ω values. There are, however, cases when 'singular frequencies' occur. The framework considers these cases which require special attention in all frequency domain methods for robustness analysis.

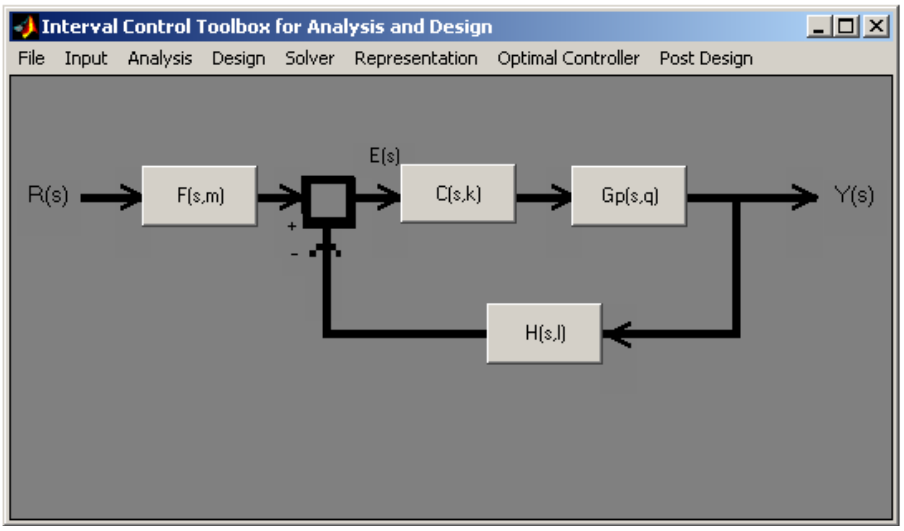

Figure 3. IRCAD framework.

5.2. Tools for Robust Design

The IRCAD framework has a powerful set of tools aimed at the design of controllers in the case of parametric control systems. Given an uncertain system, where the transfer function is defined as a family of plants q_i, from the menu design it is possible to select one or more than one specification. The specifications involve temporal and frequency domain and include: stability degree, absolute stability, resonance peak, velocity error, settling time, control effort and overshoot.

Once the specifications have been selected, each chosen specification is translated to an interval function, converting them into matrix structures and passing them to a solver (a C++ routine). The results of the solver routine, that is to say the results of the design problem, are returned on a matrix format as a set of controllers that fulfill the selected specification or the set of specifications. The resultant controllers are a set of interval values. Using the tools of Graphic User Interface from Matlab (GUI tools), the resultant matrix, which contains all the information about the valid controllers is passed to the user.

The framework allows to choose in which format we want to show the set of valid controllers. It is possible to visualize this set in a graphic way or in a numeric way. The decision often depends on the number of parameters selected for the controller. If the user wants to design a controller of two or less parameters, he can choose the graphical format, then the result will be a graph with the parameter space ranged by the parameters of the controller (k_1, k_2). In this graph the set of valid controllers are colored as a red region (**Figure 5**), the unfeasible controllers on a yellow region and the undefined regions on blue regions. If the user choose to obtain the set of valid controllers in a numeric format, then the framework shows on a table, the set of intervals for $k_1, k_2, \cdots, k_i$ that allows the system fulfill all the specifications.

5.3. Post-Design Tools

To help control engineer, IRCAD framework offers a very complete and power fulset of post-design functions. These functions allow the user, on the one hand to complete the design selecting one interval controller from the feasible controller set and in the other hand allow to verify that the selected controller achieve the specifications, without exit from application.

In order to offer a more reliable framework, the package gives the possibility to choose a criterion to select the optimal controller.

-*Optimal Controller*. The results of the robust control problem design are returned as a set of controllers that fulfill the specification or the set of specifications selected. To choose the optimal controller among this set is a task that can be not easy for the user, although the user was control engineer. To make the selection of the optimal controller from the given set of feasible controllers computed by the framework IRCAD, that offers the possibility to do it using three different criteria:

- *Minimum norm*: Given a graphical region of feasible controllers, in the parameter space delimited by the parameters of the controller K_1 and K_2, this criteria select the interval controller (K_1, K_2) that, fulfilling the specifications, has the minimum norm. This measure is calculated from the origin of the parameter space to the square defined by the interval (K_1, K_2).
- *Maximum* robustness: In this case the framework computes over each square (K_1, K_2) the value of the stability margin and selects as controller the square (K_1, K_2) that have the maximum value.
- Neighboring: Over each feasible square (K_1, K_2), the interval controller, the framework computes the number of neighbors. The square (K_1, K_2) with the maximum number of neighbors is selected.

When the user selects one of these criteria, then IRCAD computes one recommended controller following the method chosen. The framework shows the obtained controller by two ways at time: graphically and numerically. Graphically, emphasizing this controller with a dark color on the graphic, that represents the set of feasible controllers (**Figure 6**), and Numerically, where is opened a window with the recommended values for the controller.

The recommended value for the controller can be introduced on the controller box as an entry, or it can be introduced automatically as following is explained.

-*Input optimal controller*. The framework allows that the parameters of the commended controller obtained using one of the techniques exposed (minimum norm, maximum robustness or nearest neighbor) can directly be input. This automatic entry of the parameters of the controller allows to execute post-design tasks without exit from the framework.

-*Post design tools*. It is not necessary to go to any simulation environment to verify the selection, because the

selected controller can be simulated using the same framework IRCAD.
All these features give to the user a friendly environment to work in the case of robust control problems, with the security that he has obtained guarantee solutions on the results.
The mainly tools of this block are concerning to the facilities to have tools to check that a designed controller fulfill a set of specifications.

6. Illustrative Example

To illustrate how the solver created manages the data, an example suggested by [22] and later studied by [23] is used. The aim of the example is tuning a PI controller for an interval plant.

Example 1. Robust control design.

Given the transfer function of the plant:

$$G(s,\boldsymbol{q}) = \frac{q_1}{1-\frac{s}{q_2}} \tag{25}$$

where $\boldsymbol{q}$ is the vector of uncertain parameters , $\boldsymbol{q} = [q_1 \quad q_2]$; $q_1 = q_2 = [0.8, 1.25]$ and the PI controller is described by:

$$C(s,\boldsymbol{k}) = \frac{k_1\left(1+\frac{s}{k_2}\right)}{s} \tag{26}$$

where $\boldsymbol{k}$ is the vector of the design parameters, $\boldsymbol{k} = [k_1, k_2]$.

The design aim is to find the set $\boldsymbol{K}$ of controller parameters that fulfill the following performance specifications:

1. Absolute stability.
2. Velocity error lower than 2%.
3. Resonance peak lower than 3 dB.
4. Control effort lower than 20.

Given the Equation (26), the toolbox IRCAD manage each one of these specifications transforming them via the Extended Symbolic Toolbox into a set of polynomial interval functions [Equations (27)-(30)]. Matlab send these functions to the solver, transforming it previously on a matrix structure. This transformation is automatically done by IRCAD toolbox, thus it is transparent to the user. The role of the solver is to test the positivity of these functions over the parameter space, which is determined by the controller parameters. In this example the parameter space is determined by k_1 and k_2. IRCAD allows to present the results of the solver in two formats :

-Graphically.

-Set of intervals.

For this example the parameter space to be checked is limited giving the initial interval values to the parameters of the controller as $k_1 = [-200, 0]$ and $k_1 = [0, 10]$. In order to work with positive intervals the change $\overline{k_1} = -k_1$ has been made.

As is exposed at Section 2, in the subsection dedicated to robust control problems of design, to test if one specification or a set of specifications are fulfilled it is enough to test if a function or a set of functions are positive. In our example it is enough to test the positivity of the following functions:

1) For the specification of absolute stability the function obtained is:

$$f_1 = -k_2 + q_1\overline{k_1} \tag{27}$$

2) For the specification of velocity error less than 2%:

$$f_2 = q_1\overline{k_1} - 50 \tag{28}$$

3) For the specification of resonance peak lower than 3 dB:

$$f_3 = 2k_2^2\omega_1^4 - 4k_2^2\omega_1^2\overline{k_1}q_1q_2 + \overline{k_1}^2q_1^2q_2^2k_2^2 + 2k_2^2\omega_1^2q_2^2 - 4k_2\omega_1^2q_2^2\overline{k_1}q_1 + \overline{k_1}^2q_1^2q_2^2\omega_1^2 \tag{29}$$

4) And finally for control effort lower than 20:

$$f_4 = 400k_2^2\omega_1^4 - 800k_2^2\omega_1^2\overline{k}_1 q_1 q_2 + 400\overline{k}_1^{\,2} q_1^2 q_2^2 k_2^2 + 400k_2^2\omega_1^2 q_2^2 + 400\overline{k}_1^{\,2} q_1^2 q_2^2 \omega_1^2 - \overline{k}_1^{\,2} k_2^2 q_2^2 - \overline{k}_1^{\,2}\omega_1^4 - \overline{k}_1^{\,2} k_2^2\omega_1^2 - \overline{k}_1^{\,2}\omega_1^2 q_2^2 - 800k_2\omega_1^2 q_2^2 \overline{k}_1 q_1 \tag{30}$$

The resultant feasible regions for each specification are represented on **Figure 4**.

In this example, specification f_3 and specially specification f_4 have a hard computation cost due its high number of summand components and also for the high number of plant variables q_i and controller variables k_i that they involve.

Although this, when IRCAD compute the four specifications at the same time the cost of computation is significantly reduced because some regions are discarded on the specification of lost cost f_1 and f_2.

Using IRCAD, the problem of design a controller that fulfills the four specifications gives the graphical classification of **Figure 5**.

The graphical result of **Figure 5** corresponds to the following set of feasible intervals:

$$k_1 = [-70, -65] \quad k_2 = [2.75, 3]$$

$$k_1 = [-75, -65] \quad k_2 = [3, 3.25]$$

$$k_1 = [-80, -65] \quad k_2 = [3.25, 3.5]$$

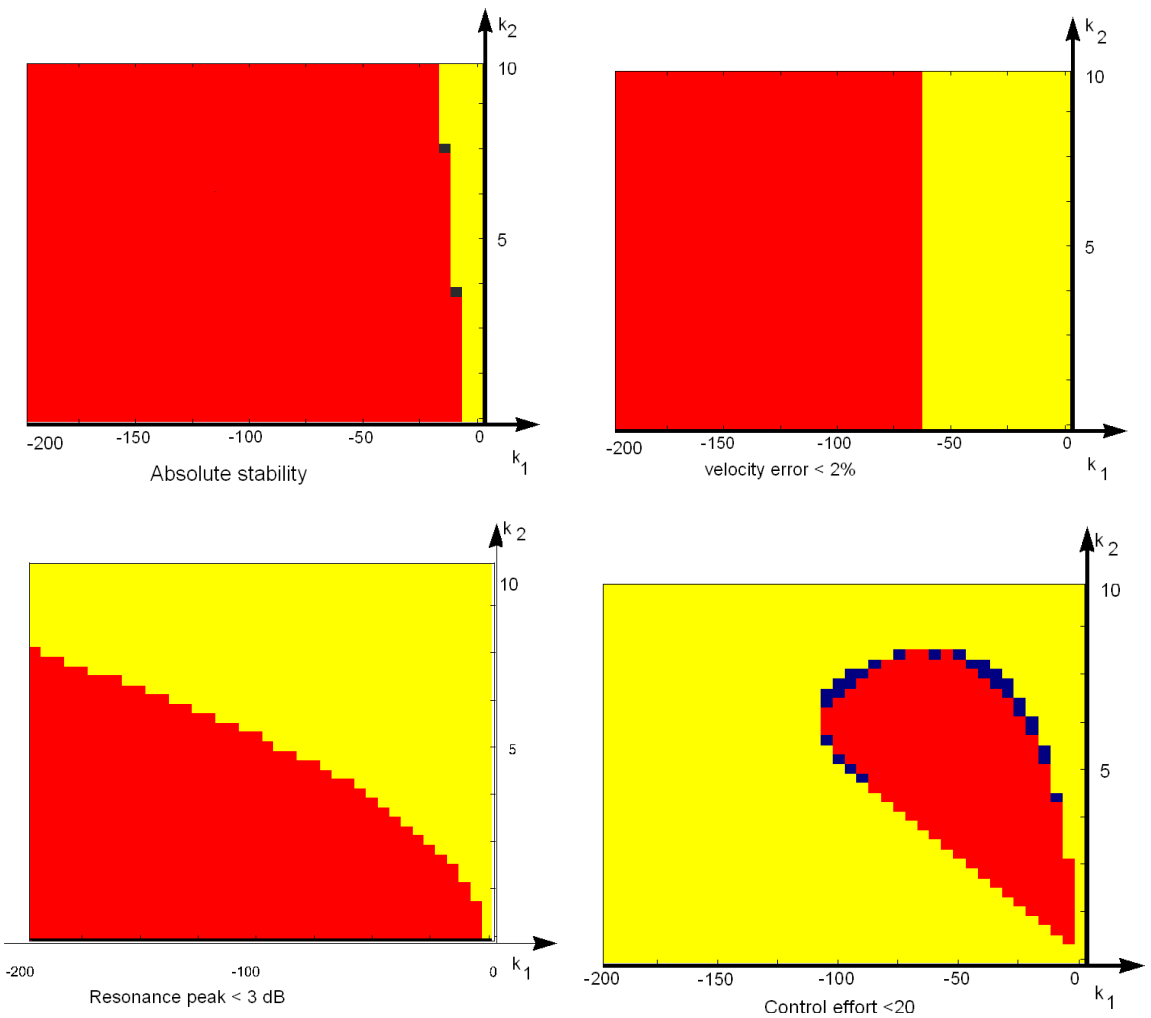

Figure 4. Feasible regions for each specification.

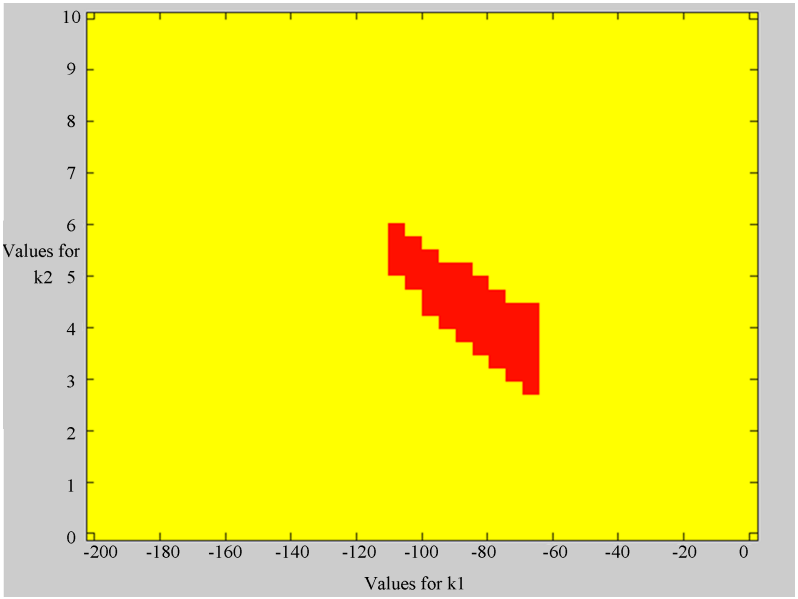

Figure 5. Feasible regions.

$$k_1 = [-85, -65] \quad k_2 = [3.5, 3.75]$$
$$k_1 = [-90, -65] \quad k_2 = [3.75, 4]$$
$$k_1 = [-95, -65] \quad k_2 = [4, 4.25]$$
$$k_1 = [-100, -65] \quad k_2 = [4.25, 4.5]$$
$$k_1 = [-100, -75] \quad k_2 = [4.5, 4.75]$$
$$k_1 = [-105, -80] \quad k_2 = [4.75, 5]$$
$$k_1 = [-110, -85] \quad k_2 = [5, 5.25]$$
$$k_1 = [-110, -95] \quad k_2 = [5.25, 5.5]$$
$$k_1 = [-110, -100] \quad k_2 = [5.5, 5.75]$$
$$k_1 = [-110, -105] \quad k_2 = [5.75, 6]$$

Finally, the solution of this robust control design example consists on the choice from this feasible region, following some criterion, of an interval pair k_1 and k_2 for the PI controller.

At this point IRCAD offers a set of post-design tools that allows to choose the optimal controller from the set of the proposed controllers on an automatic way. The framework chooses the controller following the criteria (see Section 5.3) that the user has selected.

In this example we suppose the user select the criterion of minimum norm. Then the controller suggested by the framework IRCAD is $[-67.5, 2.875]$ as is shown on **Figure 6**.

This action gives the optimal controller that fulfills the set of specifications. Using another post-design tool of the framework, the user can put this optimal controller on the closed-loop system. Finally, with this controller and this plant the framework offers some post design tools to simulate the features of the system.

7. Conclusions and Future Work

An application to solve robust control problems was designed combining symbolic tools with interval analysis. The main advantage of using interval analysis is that guaranteed results are obtained. The drawback is the ne-

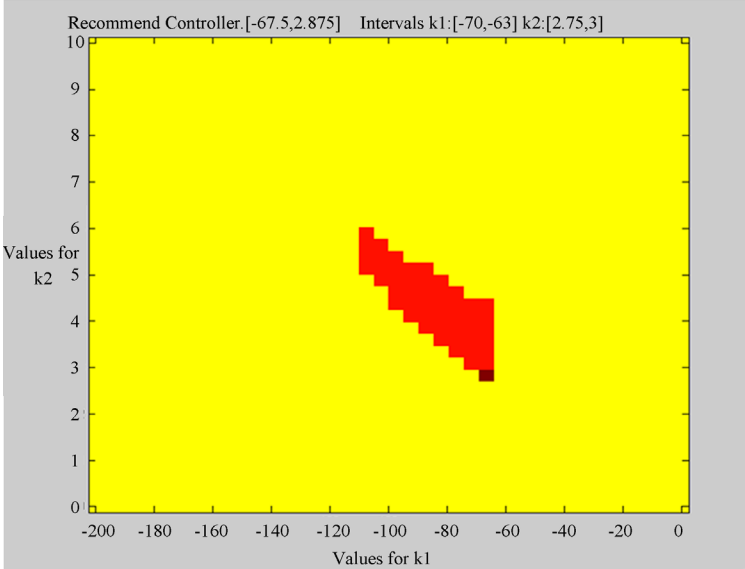

Figure 6. Minimum norm criteria.

cessity to work with a variety of data, implicit on these kinds of problems. It is necessary to manage symbolic, interval and numeric data. The manipulation of these kinds of data needs to use the fitting software in each case. This diversity can discourage control engineers because most of them don't use these data in a mixed way. The framework created, allows users to manage on an easy way with problems that involves these kinds of data in a transparent way. The result is a most reliable framework for control engineers to solve problems that involves symbolic-interval-numeric data.

References

[1] The MathWorks Inc. (1999) The Polynomial Toolbox for Matlab, v. 2.0. Technical Report. www.polyx.com

[2] Sakabe, K., Yanami, H., Anai, H. and Hara, S. (2004) A Matlab Toolbox for Robustcontrol Synthesis by Symbolic Computation. *SICE Annual Conference in Sapporo*, Hokkaido Institute of Tecnology, Japan, August 2004, 1968-1973.

[3] Sienel, W., Bunte, T. and Ackermann, J. (1996) Paradise: Parametric Robust Analysisand Design Interactive Software Environment: A Matlab-Based Robust Control Toolbox. *Proceedings of the* 1996 *IEEE International Symposium on Computer-Aided Control System Design*, Dearborn, 15-18 September 1996, 380-385.

[4] Nataraj, P.S.V. and Sardar, G. (2000) Computation of QFT Bounds for Robust Sensitivity and Gain-Phase Margins Specifications. *Journal of Dynamic Systems Measurement and Control*, **122**, 528-834.

[5] Ackermann, J. (1993) Robust Control: Systems with Uncertain Physical Parameters. Communications and Control Engineering Series. Springer-Verlag, Berlin.

[6] Hyodo, N., Hong, M., Yanami, H. and Anai, H., *et al.* (2006) Development of a Matlab Toolbox for Parametric Robust Control: New Algorithms and Functions. SICE-ICASE *International Joint Conference* 2006, Busan, 18-21 October 2006, 2856-2861.

[7] Ackermann, J. (2000) PARADISE—Parametric Robustness Analysis and Design Interactive Software Environment. Technical Report, Institute of Robotics and Mechatronics. www.op.dlr.de/FF-DRRR/paradise.

[8] Moore, R.E. (1979) Methods and Applications of Interval Analysis. In: *SIAM Studies in Applied Mathematics*, SIAM, Philadelphia.

[9] Vehí, J., Herrero, P., Sainz, M.A. and Jaulin, L. (2004) Quantified Set Inversion with Applications to Control. *IEEE International Symposium on ComputerAided Control Systems Design*, Taipei, 2-4 September 2004, 179-183.

[10] Ackermann, J. (2002) Robust Control. The Parameter Space Approach. Springer, Berlin.

[11] Frazer, R. and Duncan, W. (1929) On the Criteria for Stability of Small Motions. *Proceedings of the Royal Society A*, **124**, 642-654.

[12] Munro, N. (1999) Symbolic Methods in Control System Analysis and Design. *IEEE Control Engineering Series*, **56**, 393 p.

[13] Chapellat, H., Battacharyya, S.P and Keel, L.H. (1995) Robust Control: The Parametric Approach. In: *Information and System Science Series*, N.J. Prentice Hall, Englewood Cliffs.

[14] Garloff, J. and Graft, B. (1999) Robust Schur Stability of Polynomials with Polynomial Parameter Dependency. *Multidimensional Systems and Signal Processing*, **10**, 189-199. http://dx.doi.org/10.1023/A:1008402513296

[15] Jaulin, L. and Walter, E. (1996) Guaranteed Tuning, with Application to Robust Control and Motion Planning. *Automatica*, **32**, 1217-1221. http://dx.doi.org/10.1016/0005-1098(96)00050-7

[16] Stoorvogel, A. (1992) The H_∞ Control Problem. A State Space Approach. Prentice Hall International Series in Systems and Control Engineering, London.

[17] Salapaka, M.V., Dahleh, M. and Voulgaris, P. (1997) Mixed Objective Control Synthesis: Optimal l1/h2 Control. *SIAM*: *SIAM Journal on Control and Optimization* (*SICON*), **35**, 1672-1689.

[18] Zhou, K., Doyle, J.C. and Glover, K. (1996) Robust and Optimal Control. Prentice Hall Inc., Upper Saddle River.

[19] Gagnon, E., Pomerleau, A. and Desbiens, A. (1999) Mu-Synthesis of Robust Decentralized PI Controllers. *IEEE Proceedings Control Theory Applications*, **146**, 289-294.

[20] Vehí, J. (1998) Analysis and Design of Robust Controllers by Means of Modal Intervals. Ph.D. Thesis, Universitat de Girona, Spain. (in Catalan)

[21] SIGLA/X. (1999) Modal Intervals. In: J. Vehí and M. Sainz, Eds., *Application of Interval Analysis to Systems and Control*, Girona, Spain, 157-221.

[22] Fiorio, G., Malan, S., Milanese, M. and Taragna, M. (1993) Robust Performance Design of Fixed Structure Controllers for Systems with Uncertain Parameters. *Proceedings of the* 32*nd Conference on Decision and Control*, San Antonio, 15-17 December 1993, 3029-3031.

[23] Malan, S., Milanese, M. and Taragna, M. (1997) Robust Analysis and Design of Control Systems Using Interval Arithmetic. *Automatica*, **33**, 1363-1372. http://dx.doi.org/10.1016/S0005-1098(97)00028-9